Seminario de Investigación 17-18

Fireworks Algorithm applied to Smart Cities field & Optimizations

PhD Candidate: Clemencio Morales Lucas

PhD Director: Dr. Luis Fernando de Mingo López
Index of Contents

1. The Natural Computing Paradigm & Smart Cities
2. Fireworks Algorithm
3. Current Research
4. Investigation Results
5. Fireworks Algorithm Applications
6. Conclusions & Discussion
7. Q&A
Index of Contents

1. The Natural Computing Paradigm & Smart Cities

2. Fireworks Algorithm

3. Current Research

4. Investigation Results

5. Fireworks Algorithm Applications

6. Conclusions & Discussion
1. The Natural Computing Paradigm
1. The Natural Computing Paradigm (II)

- Natural Computing encloses three classes of algorithms:

1. Those that take inspiration from nature for the development of novel problem-solving techniques.

2. Those that are based on the use of computers to synthesize natural phenomena.

3. Those that employ natural materials, such DNA or molecules, to compute.
1. The Natural Computing Paradigm (III)

- Natural Computing spectrum:
1. The Natural Computing Paradigm (IV)

- Darwin, 1859: “Living beings have been forced to a continuous evolutive process looking for survival”

- Natural Selection & Evolution:
1. The Natural Computing Paradigm (V)

- GA: *Set of ordered instructions, that aim to evolve a population to reach a solution (improved population)*

- Genetic Algorithms (GA) have the following set of atomic elements:
1. The Natural Computing Paradigm (VI)

- Genetic Algorithms flux diagram:

- Selection: *Truncation* approach recommended:

\[
\alpha^*_T(s, T)(f_i) = s^*(f_i) = \begin{cases} 
0 & \text{if } S(f_i) \leq (1 - T)N \\
\frac{S(f_i) - (1 - T)N}{T} & \text{if } S(f_i - 1) \leq (1 - T)N < S(f_i) \\
\frac{s(f_i)}{T} & \text{else}
\end{cases}
\]
1. The Natural Computing Paradigm (VII)

- Genetic Algorithm sections for Flappy Bird Algorithm:
1. The Natural Computing Paradigm (VIII)

• PSO: Reynolds, 1987: “Technique to optimize a problem due to a meta-heuristic strategy. Iterative improvement of a candidate solution with regards to a pre-stipulated quality criteria”
1. The Natural Computing Paradigm (IX)

- Particle Swarm Optimization (PSO) pseudo-algorithm:

```java
for each (particle within S) {
    position = generateRandomValue(S[i], b_low, b_up);
    position = bestKnownPositionByParticle(S[i]);
    if (f(p) < f(g)) {
        bestGlobalPosition = position;
    }
    speed = generateRandomSpeed(b_low - b_up, b_low - b_up)
}

while(!stopCriteria) {
    for each (particle within S) {
        for each (dimension within d) {
            first_op = omega * v(i, d) + phi_p * r_p;
            second_op = (p(i, d) - x(i, d));
            third_op = phi_g * r_g * (g_d - x(i, d));
            d[i] = first_op * second_op + third_op;
        }
        Position += d[i];
        if (f(xi) < f(pi)) {
            bestParticleLocalPosition = xi;
            if (f(pi) < f(g)) {
                bestGlobalPosition = pi;
            }
        }
    }
}
return bestGlobalPosition;
```
1. The Natural Computing Paradigm (X)

http://www.realflow.com/
1. The Natural Computing Paradigm (XI)

- Dorigo & Di Caro, 1992: “Multi-agent paradigm inspired on the ants idiosyncrasy when searching for livelihood”

- Goss Experiment with *Iridomyrmex humilis* colony:
1. The Natural Computing Paradigm (XII)

- How do all the ants *know* what is the shortest path?

- **Stigmergy**: Collaboration protocol where communication is made due to the accumulation of objects, such as pheromones

\[
p_{ij}^k \begin{cases} 
\tau_{ij} & \text{if } j \in N_i \\
0 & \text{if } j \notin N_i 
\end{cases}
\]
1. The Natural Computing Paradigm (XIII)

- Urban cores massification ➔ Urgent **improvement** needed
Index of Contents

1. The Natural Computing Paradigm & Smart Cities

2. Fireworks Algorithm

3. Current Research

4. Investigation Results

5. Fireworks Algorithm Applications

6. Conclusions & Discussion
2. Fireworks Algorithm

• FWA: Incremental and iterative search process in a huge solution space.
2. Fireworks Algorithm (II)

Select $N$ initial locations

Set off $N$ fireworks at $N$ locations

Obtain sparks location

Evaluate quality of the locations

Select $N$ locations

Optimal location found

Yes

No

End
2. Fireworks Algorithm (III)

- **FWA versus CPSO** (*Combinatorial Particle Swarm Optimization*) & **SPSO** (*Standard Particle Swarm Optimization*) → 8 Benchmark $f(x)$ over 20 runs:
2. Fireworks Algorithm (IV)

- $\bar{x}$ and $\sigma_x$ for FWA, CPSO and SPSO on 9 benchmark $f(x)$ → 20 independent runs of 10000 $f(x)$ evaluations:

<table>
<thead>
<tr>
<th>Function</th>
<th>FA’s mean (StD)</th>
<th>CPSO’s mean (StD)</th>
<th>SPSO’s mean (StD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphere</td>
<td>0.000000 (0.000000)</td>
<td>11857.425781 (3305.973067)</td>
<td>24919.099609 (3383.241523)</td>
</tr>
<tr>
<td>Rosenbrock</td>
<td>19.38330 (11.94373)</td>
<td>2750997504.000000 (1741747548.420642)</td>
<td>5571942400.000000 (960421617.568024)</td>
</tr>
<tr>
<td>Rastrigrin</td>
<td>0.000000 (0.000000)</td>
<td>10940.148438 (3663.484331)</td>
<td>24013.001953 (4246.961530)</td>
</tr>
<tr>
<td>Griewank</td>
<td>0.000000 (0.000000)</td>
<td>3.457273 (0.911027)</td>
<td>7.125976 (0.965788)</td>
</tr>
<tr>
<td>Ellipse</td>
<td>0.000000 (0.000000)</td>
<td>2493945.500000 (1199024.648305)</td>
<td>5305106.500000 (1117954.409340)</td>
</tr>
<tr>
<td>Cigar</td>
<td>0.000000 (0.000000)</td>
<td>122527168.000000 (28596381.089661)</td>
<td>149600864.000000 (13093322.778560)</td>
</tr>
<tr>
<td>Tablet</td>
<td>0.000000 (0.000000)</td>
<td>15595.107422 (8086.792234)</td>
<td>42547.488281 (8232.221882)</td>
</tr>
<tr>
<td>Schwefel</td>
<td>4.353733 (1.479332)</td>
<td>8775860.000000 (1217609.288290)</td>
<td>6743699.000000 (597770.084232)</td>
</tr>
<tr>
<td>Ackley</td>
<td>0.000000 (0.000000)</td>
<td>15.907665 (1.196082)</td>
<td>18.423347 (0.503372)</td>
</tr>
</tbody>
</table>
Index of Contents

1. The Natural Computing Paradigm & Smart Cities

2. Fireworks Algorithm

3. Current Research

4. Investigation Results

5. Fireworks Algorithm Applications

6. Conclusions & Discussion
3. Current Research

- Currently centered on Fireworks Algorithm
- Eleven Benchmarking functions for testing algorithm’s performance

```java
private double ackleyFunction(final double[] inputValues) {
    double aux = 0, result = 0;
    for (int i = 0; i < inputValues.length; i++) {
        result += Math.pow(inputValues[i], DOUBLE_SQUARE_VALUE);
        aux += Math.cos(SQUARE_VALUE * Math.PI * inputValues[i]);
    }
    result = ACKLEY_FUNCTION_NEGATIVE_MULTIPLIER * Math.sqrt(result / inputValues.length);
    result = ACKLEY_FUNCTION_LOWER_BOUND * Math.exp(result);
    aux = -Math.exp(aux / inputValues.length);
    result += ACKLEY_FUNCTION_UPPER_BOUND + Math.exp(DOUBLE_IDENTITY_VALUE) + aux;
    return result;
}
```

\[ f(x) = -a \exp \left( -b \sqrt{\frac{1}{d} \sum_{i=1}^{d} x_i^2} \right) - \exp \left( \frac{1}{d} \sum_{i=1}^{d} \cos(c x_i) \right) + a + \exp(1) \]
3. Current Research (II)

• Pre-initialization for Spark generation in FWA

• Hardware warm-up phase for accurate results

• Executing FWA to optimize mathematical functions

• Gaussian Explode (adding a random value from a Gaussian Distribution to each spark’s direction a new area of interest) for FWA Sparks
3. Current Research (III)

- Classic FWA & Pre-initialized FWA are launched

```java
private void launchFWAForAllParameters() {
    for (int i = 1; i <= BenchmarkFunctionConstants.NUMBER_OF_FUNCTIONS; i++) {
        System.out.print(FITNESS + i + SEPARATOR);
        maximumBound = new double[availableDimensions[i - 1]];
        minimumBound = new double[availableDimensions[i - 1]];
        for (int j = 0; j < maximumBound.length; j++) {
            maximumBound[j] = availableBounds[i - 1];
            minimumBound[j] = -availableBounds[i - 1];
        }
        for (int k = 0; k < shiftIndex.length; k++) {
            BenchmarkFunction benchmarkFunction = new BenchmarkFunction();
            benchmarkFunction.setIndexAndShift(i, availableBounds[i - 1] * shiftIndex[k]);
            double avg = 0;
            for (int t = 0; t < NUMBER_OF_ITERATIONS; t++) {
                FireworkAlgorithm fireworkAlgorithm = new FireworkAlgorithm(
                    LOCATIONS_NUMBER, NUMBER_OF_SPARKS,
                    LOW_BOUND_NUMBER, HIGH_BOUND_NUMBER, MAXIMUM_AMPLITUDE_VALUE, GAUSSIAN_SPARKS_VALUE,
                    maximumBound, minimumBound, filePath, benchmarkFunction);
                avg += fireworkAlgorithm.launch();
            }
            avg /= NUMBER_OF_ITERATIONS;
            System.out.print(NEW_LINE + avg);
        }
    }
    System.out.println();
}
```
Index of Contents

1. The Natural Computing Paradigm & Smart Cities
2. Fireworks Algorithm
3. Current Research
4. Investigation Results
5. Fireworks Algorithm Applications
6. Conclusions & Discussion
4. Investigation Results

- Classic FWA for 6 Shifts (iterations) in 12 Fitness calculations:
4. Investigation Results (II)

- Pre-Initialized FWA for 6 Shifts (iterations) in 12 Fitness calculations:

| Fitness 1 | 78630.9791875285 | 104210.02956264642 | 119887.85049835381 | 131573.81130339826 | 172176.6628978758 | 197438.2713592046 | 337108.7771787091 |
| Fitness 2 | 1378096.5105605469 | 1217939.50605585406 | 1515021.1052623462 | 2045832.3654442176 | 2619227.542479729 | 2532139.6595763036 | 4613281.2906126002 |
| Fitness 3 | 2.5561141162213668E8 | 4.0098386511205182E8 | 6.756625873966776E8 | 8.625029362234106E8 | 1.207145263687446E9 | 1.6962818423289623E9 | 4.792003363670427E9 |
| Fitness 5 | 792.1730450304727 | 840.2933710982547 | 832.827144019712 | 1438.066596475276 | 1602.73331579155 | 1573.857289035941 | 2749.558291524064 |
| Fitness 6 | 358.9404765830502 | 400.70684762828 | 422.3786690385915 | 445.6142116617796 | 518.40423213165 | 621.296240306615 | 811.9978132870777 |
| Fitness 7 | 1.449632910076408E9 | 2.4864635165290046E9 | 3.04512423193476E9 | 5.103656020653241E9 | 7.962119953095451E9 | 9.430240181799067E8 | 3.251091837974155E10 |
| Fitness 8 | 186.15021505013607 | 389.830708789191 | 827.439931058842 | 2085.056122119717 | 4147.59493531317 | 12478.75370758119 | 3006.87607513805 |
| Fitness 9 | 435.11230116609016 | 359.942295346167 | 694.94525109795041 | 1312.60778705835 | 2382.45972232532 | 6994.74443173751 | 14924.330772122154 |
| Fitness 10 | 0.4916679020876662 | 0.4941630551640005 | 0.49552214061975407 | 0.497132833135547 | 0.4985155942171041 | 0.49950467945420535 | 0.499715291619254 |
| Fitness 11 | 1296.74881125524 | 2142.182410697925 | 2043.006188501836 | 2943.41830682275 | 3284.726674467795 | 5165.509516610583 | 8275.061493082163 |
| Fitness 12 | 3986381.226555627 | 5776834.4065739005 | 7990955.610401971 | 6501259.764193256 | 9187848.875311902 | 1.525565354759997E7 | 4.1041761097279236E7 |

Time elapsed for improved algorithm (ns): 6095082401
4. Investigation Results (III)

- Ahmdal’s Law \(\rightarrow\) Non-Improved Time

\[
\text{Improvement (\%) = } \frac{\text{Improved Time}}{\text{Non-Improved Time}}
\]

---

*Time elapsed for improved algorithm (ns): 6095082401*

---

Ahmdal's Law = Non-Improved time / Improved Time = \(\frac{7572587531}{6095082401} = 124.24093773953886\)

The improved algorithm is a \(24.24093773953886\%\) faster
4. Investigation Results (IV)

• Next investigation path: Searching for a synergy between CPSO & FWA.

• In CPSO, Global Best \((g\text{-best})\) and Personal Best \((p\text{-best})\) are taken into computation.
4. Investigation Results (V)

- Global Best \((g\text{-best})\) and Personal Best \((p\text{-best})\) just take swarm-related data.

- In Nature, the environment plays a central role → **Extrapolation to CPSO & FWA adding E-best:**

- A spark explosion success determines next spark’s exploding point:
Index of Contents

1. The Natural Computing Paradigm & Smart Cities
2. Fireworks Algorithm
3. Current Research
4. Investigation Results
5. Fireworks Algorithm Applications
6. Conclusions & Discussion
FWA+ACO (II)
5. Feasible application for Smart City rural area

Mountain Control Center/Wooded area
5. Feasible application for Smart City (II)

- **Travelling Salesman Problem (TSP):** "Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?"

- NP-hard problem in combinatorial optimization
5. Feasible application for Smart City (III)

- Travelling Salesman Problem (TSP) + Ant Colony Optimization (ACO)
- **Massive computation 😞**
- Ants can be used to trim the search tree down 😊
5. Feasible application for Smart City (IV)

- System for endowing intelligence to the UPTS in development:
5. Feasible application for Smart City (V)

- **Genetic Algorithms** to *evolve* a route. Mobile application to:
  - **Inform users about backup routes** in case of systems breakdown
  - **Calculate route** between two points, even if there are blocked sections due to failure
  - **User super-pheromone** for statistical use
5. Feasible application for Smart Space (VI)

- Search for Extraterrestrial Intelligence (SETI)
6. Conclusions & Discussion

- **Citizens** can be seen as *particles* among a huge *swarm*. Natural Computing algorithms are highly mixable.

- PhD investigation efforts currently centered in the **Fireworks Algorithm** and an optimization has been reached
  - FWA with pre-initialization is a 24% faster :-(
  - Investigating synergy between FWA and ACO
6. Conclusions & Discussion

**Base**
- The Natural Computation paradigm has been investigated, as well as its application in Smart Cities spectrum

**Investigation**
- A FWA optimization based on pre-initialization scheme has been given, as well as a system for finding & rescuing lost people in rural areas (FWA+ACO)

**Development**
- FWA speed has been increased by 24% and the synergy between FWA+ACO looks promising.
Seminario de Investigación 17-18

Fireworks Algorithm applied to Smart Cities field & Optimizations

PhD Candidate: Clemencio Morales Lucas
Contact: mail@clemenciomorales.com

PhD Director: Dr. Luis Fernando de Mingo López